Wikipedia & Chatting 프로토콜 설계

201904014 성민창

<Service>

[wikipedia]: 클라이언트는 서버에 다양한 정보를 저장하고 불러오며, 변경하고 삭제할 수 있다. 그리고 이러한 정보는 모든 클라이언트가 공유하여 위키피디아처럼 사용할 수 있다.

[chatting]: 위키피디아 서비스 외에 서버에 연결된 클라이언트 간의 1:1 텍스트 통신이 가능하다.

* 통신은 문자 단위로 이루어지며 파일 전송은 서비스하지 않는다.

<Message format>

code(ID)	**/**	**fromID**	**CRLF**				
서비스 유형	구분자	송신자 ID	구분자				

-ID 통신은 동기적으로 수행된다. (AtomicBoolean 사용)

code(WIKI)	**/**	**CRUD**	**/**	**body**	**(/)**	**(+body)**	**CRLF**
서비스 유형	구분자	-CREATE -READ -UPDATE -DELETE	구분자	-UnitID: READ, UPDATE, DELEETE -Message: CREATE	구분자	-Message: UPDATE	구분자

-wiki 통신은 동기적으로 수행된다. (ReentrantLock 사용)

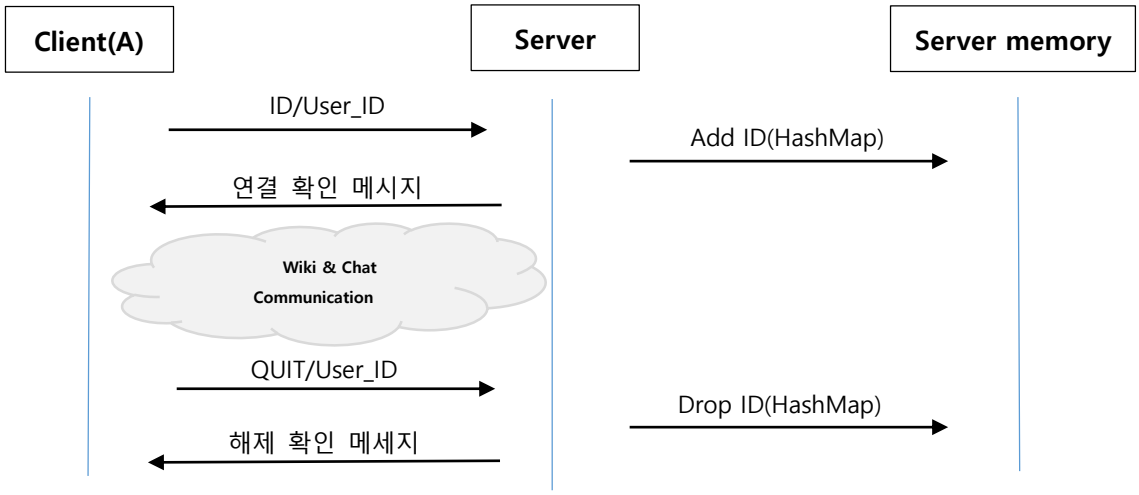
code(CHAT)	**/**	**fromID**	**/**	**toID**	**/**	**body**	**CRLF**
서비스 유형	구분자	송신자 ID	구분자	수신자 ID	구분자	본문	구분자

-chat 통신은 비동기적으로 수행된다.

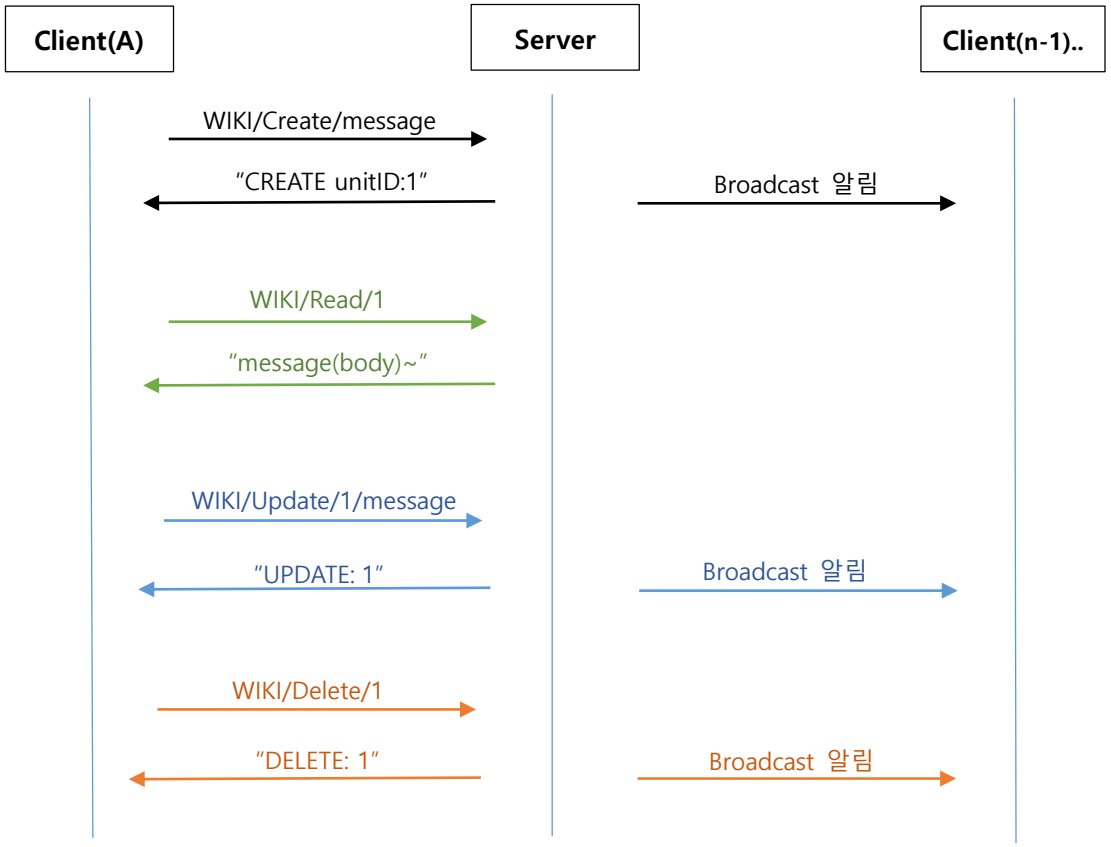
code(QUIT)	**/**	**FromID**	**CRLF**				
서비스 유형	구분자	송신자 ID	구분자				

-quit 는 연결을 종료하며 비동기적으로 수행된다.

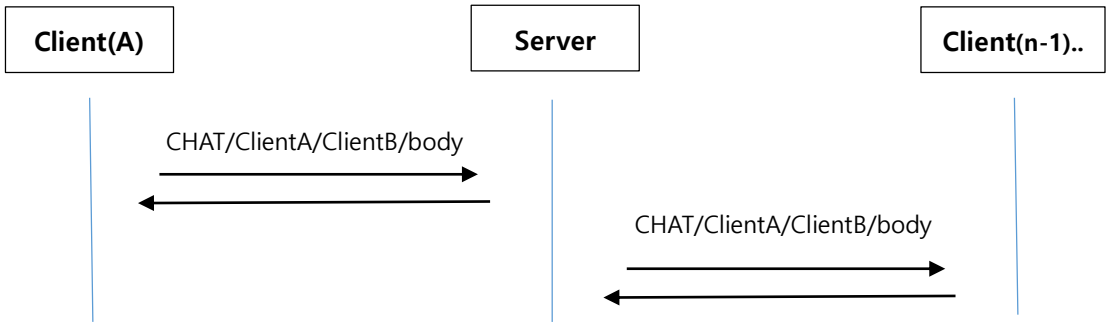
<Sequence Diagram – ID Management>



<Sequence Diagram - Wiki Communication>



<Sequence Diagram - Chat Communication>



<Feature point>

[WIKI 문서의 자료구조]

새 WIKI 문서가 생성될 때마다 1 씩 증가된 unitID 를 할당받는데 이렇게 되면 WIKI 문서가 삭제되었을 때 더 이상 사용되지 않는 unitID 가 생겨 unitID 가 불필요하게 늘어나는 문제가 있다.

이를 방지하기 위해 문서가 삭제되면 해당 문서가 가지고 있던 unitID 를 spare 라는 queue 구조에 저장하여 다음에 생성되는 WIKI 문서의 unitID 로 재활용할 수 있도록 자료구조를 설계했다.

[동기화 알고리즘]

Java.util.concurrent.locks 라이브러리를 활용하여 다수의 클라이언트가 WIKI 서비스에 동시 접근할 때 생길 수 있는 동시성 문제를 차단하고자 하였다.

Java.util.concurrent.atomic.AtomicBoolean 라이브러리를 활용하여 클라이언트가 자신의 fromID 를 서버에 전송했을 때 서브 스레드에서 서버의 응답메세지를 받기 전에 메인 스레드에서 다음 코드를 실행시키지 못하도록 while(lock.get());루프로 서브 스레드와 메인 스레드를 동기화시켰다.

<Implement method>

1.서버 프로그램을 먼저 실행시킨 뒤, 클라이언트 프로그램을 실행시킨다.

2.클라이언트 프로세스에서 채팅에 사용하고자하는 ID 를 입력한다.

3.usage 매뉴얼에 따라 메시지를 입력한다. 하나의 메시지는 'enter'를 기준으로 서버에 전송된다.